# THE COST OF PROTECTION MEASURES IN TACTICAL NETWORKS[*]

Brian J. Matt[†]
McAfee Research[‡]
Rockville, MD

## 1. INTRODUCTION

The security of the Object Force and Future Combat Systems communication is dependent on the Army research community's ability to solve difficult problems in ad hoc routing security. Solving these problems will require (in part) the development of innovative cryptographic methods for protecting routing messages and other communications in wireless ad hoc networks, including strong, efficient methods for authentication.

Traditional methods such as digital signatures have significant performance costs, even so routing security researchers have used them to protect routing control messages. Recently, researchers, focusing mainly on commercial networks, have adopted a number of well known lighter weight cryptographic techniques, such as one-time digital signatures (OTDS) and authentication trees, to construct new techniques for faster routing packet authentication and integrity protection. When are these new techniques the correct approach for protecting tactical networks, networks in which the performance characteristics of the devices and protocols can be quite different from commercial counterparts?

Recently, new techniques for digital signatures have appeared including new techniques for identity-based signatures and new techniques for short signatures. Do these techniques offer advantages to tactical network protocol designers? Are the advantages limited to very low data rate channels or do these techniques have broader applicability?

In this paper we describe some results from our search for answers to these questions. We present a simple delay performance model and results of our analysis of these new techniques vs. traditional signature techniques in this model.[1] Our results show that over the wide range of performance exhibited by tactical systems and interesting scenarios, no technique provides the best performance. However, in situations where medium to low bandwidth channels are used[2] it is typically the case that some novel, computationally efficient authentication techniques are outperformed by traditional signatures, and these traditional signatures can be outperformed by recently developed novel, communion efficient, but computationally intensive techniques.

In the remainder of this paper we summarize the model used in this study in Section 2, and provide a brief overview of the techniques we studied in Section 3. In Section 4 we discuss our results and in Section 5 we discuss our conclusions and areas for future work.

## 2. THE MODEL

We use a simple communication model for comparing the delay caused by the uses of different authentication techniques. We assume a Carrier Sense Multiple Access (CSMA) channel with variable length packets, where the addition of different authenticators does not result in the generation of additional packets for the message. We also assume that the delay that a packet experiences moving from node to node is fixed with respect to packet size, except for the packet transmission delay. We base our performance analysis of the protocols at a particular data rate on the rate being *realistic*. However, the data rates for various radios that appear in the paper are typically *raw* data rates. We further assume that the bit-error-rate of the channel is sufficiently low so that the variations in packet size, due to the choice of authentication technique, do not have a noticeable impact on the performance of the technique. We note that including the creation of additional packets, due to the inclusion of an authenticator, would add a disproportionate penalty to the OTDS and authentication tree techniques, and to a lesser extent to RSA signatures. The average packet delay, preamble size, header size, and max packet size, would be factors affecting these results.

We model the computational costs of the authentication algorithms using a 933/400 MHz Pentium III (Processor-M Ultra Low Voltage) at both clock speeds. We also reduced the performance of the processor (by

[*] Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

[†] The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

[‡] Formerly Network Associates Laboratories

[1] Our focus is on situations where (multiple) use of a traditional *point-to-point* authentication technique is inappropriate.

[2] Or the channel performance is significantly degraded.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

| | | |
|---|---|---|
| 1. REPORT DATE<br>**00 DEC 2004** | 2. REPORT TYPE<br>**N/A** | 3. DATES COVERED<br>**-** |

4. TITLE AND SUBTITLE
**The Cost Of Protection Measures In Tactical Networks**

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**McAfee Research Rockville, MD**

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release, distribution unlimited**

13. SUPPLEMENTARY NOTES
**See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2005 in Orlando, Florida. , The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | **UU** | **8** | |
| **unclassified** | **unclassified** | **unclassified** | | | |

factors of 4, 10, 40, etc.) to better understand how sensitive the authentication techniques are to changes in processor performance. We assume that reduced availability of the processor (special purpose processoror other computational device) for other tasks due to authenticator generation and verification does not have a significant impact on the system overall, and that the processor is available for use for authentication tasks with constant or minimal delay.[3]

The increased communication delay caused by each authentication scheme is measured in bits. Computation costs are converted into bits, i.e., the cost of a 35 mSec computation on a system with a 40 Kbps channel is 1400 "bits" while the same computation on the same processor at the same clock speed on a system with a 12,000 Kbps channel is 4.2 x $10^5$ "bits." The computation costs of computationally intensive algorithms are based on performance measurements using a 1.0 GHz Pentium III (Barreto et al., 2002; Barret, 2003). The computational costs of the lightweight algorithms are estimated. The algorithms' computational costs are dominated by their use of hash function and MACs. We use the measured performance hash functions using MD-5 and SHA-1 and HMACs as the basis of these estimates. Issues with the security of hash functions are beyond the scope of this paper.

## 3. AUTHENTICATION PROTOCOLS

We can describe a digital signature and other authentication protocols in a general way has having an (re)initialization phase, and an operational phase. For our purposes it is sufficient to say that the initialization phase involves the secure distribution (frequently by a message sender) of information necessary for verifying authenticators generated by the sender to a set of potential message receivers. The potential receivers verify the authenticity and freshness of the information and may take additional steps to prepare for verifing traffic from the sender. A familar example of this phase would be the distribution of a certificate authenticating a public digital signature verification key by a message sender or by using a certificate directory.

During the operational phase the sender generates authentication information for a message (perhaps using a pre-computation step), and distributes the message and authenticator to the recipients. The verifiers use the message, the authenticator, and current state to verify or reject the authenticity of the message. In some authentication schemes the operational phase is combined with re-initialization, e.g., the Independent One-time Signature Protocol described in Section 3.3. In delayed verification schemes (Perrig et al., 2002) the operational phase is divided into two parts, during the first part messages and authenticators are distributed, but the receiver do not have sufficient information to verify these messages. Message verification must be delayed until a time interval ends and an additional message, which provides the additional infromation need to verify the messages, is distributed during the second part. We do not study such delayed schemes in this paper, however, the TESLA with Instant Key Disclosure scheme, see Section 3.4 is related to these schemes.

### 3.1 Digital signatures

The concept of a digital signature and the use of certificates to distribute public keys for signature verification are well known and we do not discuss the details of specific schemes here.[4] The schemes used in this analysis are the Rivest, Shamir and Adleman (RSA) algorithm (Rivest et al., 1978), Digital Signature Algorithm (DSA) and Elliptic Curve DSA (ECSDA) using $F_p$ or $F_{2^n}$ (FIPS 186-2, 2001). We also examined a recent advance in short signatures, the Boneh, Lynn, Shacham (BLS) scheme (Boneh et al., 2001. This new technique is based on the Weil pairing, a mathematical technique that along with the Tate pairing, has recently been widely used in cryptographic research. The computational performance of all of these schemes is significant, but the message sizes, except for RSA, are relatively small. For a level of security similar to that 1024 bit RSA, the costs of these schemes is given by Table 1 for a 1.0 GHz Pentium III.

Table 1: Public-key Signature Schemes

| Algorithm | Generation Time (mSec) | Verification Time (mSec) | Bandwidth (bits) |
|---|---|---|---|
| RSA | 7.9 | 0.4 | 1024 |
| DSA | 4.1 | 4.9 | 320 |
| ECDSA $F_{2^{160}}$ | 5.7 | 7.2 | 320 |
| ECDSA $F_p$ | 4.0 | 5.2 | 320 |
| BLS $F_{3^{97}}$ | 3.5 | 23.0 | 170 |

### 3.2 Identity-based digital signatures

In an identity-based digital signature scheme (Shamir, 1985) the public key used for signature verification can be generated by any party from the public system parameters of a Trusted Authority (TA), and the identifier of the signer. The identifier can include not only the name of the signer, but administrative information such as the validity

[3] We consider both the situation where the processor (or similarly performing hardware) is shared with transceiver system and when the processor in not shared. In certain military systems, hardware subsystems used for communications are suitable and available (on a limited basis) for use performing cryptographic tasks.

[4] In this work we only considered signature schemes with appendix, i.e., did not consider signature schemes with (partial) message recovery.

period of the corresponding public key. The authority generates private keys for a signer from the same identifier used to generate the signer's public key, and the private system parameters. The private key of the signer is sent by the TA to the signer over a private and authenticated channel. Prior to verifying signatures, verifiers must obtain the authentic public system parameters of the authority, and information on how to construct identifiers.

During the initialization phase the signer has to distribute only that information that is not part of a normal message, and is not known to the verifiers, but which is needed to construct the sender's identifier. The rules for identifier construction can be designed to eliminate the need for an initialization phase and avoid adding significant overhead to ordinary messages. For example, the authority could generate keys with six month long validity periods. Each period is either January 1st through June 30th or July 1st through December 31st. The verifier will typically perfrom the verifcation step for a new operational message using the current validity period, we include a flag in our messages to aid verification during the transition between periods.

During the operational phase the signer uses its private key to sign a message and distributes the message and signature. A receiver can use the public system parameters and other global knowledge, along with information about the sender taken from the message, message header, etc., and determine the public key of the sender. The receiver then verifies that the signature for the message was computed by the signer.

In our analysis we use the identity-based signature scheme of Cha and Cheon (Cha and Cheon, 2003) which is based on the Weil or Tate pairings. The values for the costs of this scheme used in this study are: generation 4.0 mSec; verification 24 mSec; and signature size 340 bits on a a 1.0 GHz Pentium III.

### 3.3 Approaches based on OTDS

One-time digital signatures (Lamport, 1979) are mechanisms that can use a public key to sign at most one message; otherwise the signature can be forged. A new public key is needed for each signature by the same signer. To be practical for our anticipated applications the signature process must be computationally and communication-efficient, and the OTDS mechanism must be extended so that multiple public keys can be efficiently distributed. In (Zhang, 1998), two techniques for protecting routing messages based on one-time signatures were presented: the Chained One-time Signature Protocol (COSP); and the Independent One-time Signature Protocol (IOSP).

In COSP during the initialization phase, a signer generates a random set of $n$ secrets $x_j,\ j = 1,...,n$. For each secret, $x_j$, the signer uses the hash function $h$ to compute a $k$ length hash chain $h^k(x_j)$, and the set hashed values $h^k(x_j),\ j = 1,...,n$, is the COSP public key of the signer.

This public key may be signed using a signature scheme from Sections 3.1 or 3.2, and the public key and signature (along with a certificate if necessary) are distributed. The value $n$ is the size of the output of a hash function $\hat{h}$ plus the size of a counter. In (Zhang, 1998) MD5 and SHA-1 were suggested for the hash functions. In a system that uses SHA-1 for both hash function the size of the COSP public key for a reasonable sized counter is over 28 hundred bits.

To sign a message $M$ the message is hashed, and the hash is concatenated with and counter (with a value $l$). For each bit of the result, if bit $j$ is set to one, the corresponding value $h^l(x_j)$ is included in the signature. The signature is $l \ || \ \{ \ sub\_set\_of(\{ \ h^l(x_j), \ j = 1,...,n \ \}) \ \}$. The average size of a signature is over 14 hundred bits (using SHA-1). The counter, which is initially zero, is incremented for the next signature.

To verify a message the receiver hashes the message and determines for the hash and the counter whether the values from the signature $\{v_j,\ j = 1,...,n'\}$, where $n' \leq n$ are consistent with the COSP public key of the sender. This process requires computing the value $h^{(k-l)}(v_j)$ for each $v_j$ and comparing theses values with the values in the appropriate positions in the public key.

In COSP signatures with higher counter values contain information that can be used to forge signatures with lower counter values, in a delay and forge attack (Hauser et al., 1997). The approach to addressing this problem used in COSP, discussed in (Hauser et al., 1997), requires synchronization between sender and receiver. The sender has to sign messages at a fixed time interval $T$ (skipping intervals is permitted), and the receiver's clocks have to be synchronized with the sender. The interval $T$ between allowed signings must be long enough for a message to propagate through the network to reach the intended receivers. If a receiver misses a message it may verify later messages without re-initializing.

In IOSP during the initialization phase a signer generates a random set of $n$ secrets $\{x_j,\ j = 1,...,n\}$. The ISOP public key is $P = h( \ h(x_1) \ || \ \cdots \ || \ h(x_n) \ )$. This public key may be signed using a signature scheme from Sections 3.1 or 3.2 and the combination (along with a certificate if necessary) is distributed. The value $n$ is the size of the output of a hash function $\hat{h}$ plus the size of a counter. In a system that uses SHA-1 for $h$, the size of the IOSP public key is 160 bits.

To sign a message a new IOSP public key $P'$ is created using the technique described in the proceeding paragraph, then message $M$ and the new public key are concatenated and hashed, and the hash is concatenated with the value of a counter (with a value $l$). For each bit of the result, if bit $j$ is set to zero, the corresponding value $h(x_j)$ is included in the signature. If the bit is one, the value $x_j$ is included in the signature. The signature is $l \mid\mid \{ \ v_j = ( \ h(x_j) \ or \ x_j \ ), \ j = 1, ..., n \ \}$. A typical average size for such a signature is over 28 hundred bits (using SHA-1). The counter is incremented for each new signature. The per message authentication communication overhead is the length of signature and the public key $P'$.

To verify a message the receiver hashes the message concatenated with the new public key, and concatenates the result with the counter, producing the value $g$. The verifier determines for $g$ whether the values $\{v_j, \ j = 1, ..., n \ \}$, are consistent with the ISOP public key of the sender from the previous message. This step involves computing $h(v_j)$ as necessary (for those bits in $g$ that are set to one), computing a value $V$ from the appropriate concatenation of values $v_j$ and $h(v_j)$, and comparing $V$ with the previously distributed $P$. If a receiver misses a message and a public key update, the sender and receiver must *re-initialize*.

### 3.4 Authentication trees

Authentication trees (Merkle, 1980) are mechanisms that enable the disclosure of a set of public value, in any order, with verifiable authenticity. In a binary authentication tree each leaf node $n_i$ is assigned a value $K_i$ and the hash $h_i = h(K_i)$. Each interior node $n_k$, with child nodes $n_i$ and $n_j$, has the value $h_k = h(h_i \mid\mid h_j)$. In order to distribute the tree, the root value is distributed using an authenticated channel. The value $K_i$ can then be authenticated by disclosing the values $h_i$ for the sibling node for each node in the path from the leaf node $n_i$ to the root. We will call these $h_i$ values for the leaf $n_i$ as $Apath(K_i)$.

In (Hu et al., 2003) a protocol called TESLA[5] with Instant Key Disclosure (TIK) is presented which is based on authentication trees. In TIK a sender broadcasts messages protected by a Message Authentication Code (e.g., a HMAC) and then discloses the session (single message) key for the MAC in the same message. This is somewhat similar to the design of the delayed-authentication schemes, but by using a seperate key per message it is possible to disclose the key in the same message.

During the initialization phase, the sender generates a binary authentication tree and can distribute the tree by signing the value for the root of the tree (160 bits using SHA-1) using a signature scheme from Sections 3.1 or 3.2 and the combination (along with a certificate if necessary) is distributed.[6]

During the operational phase, a key (a value $K_i$ from a tree) is used to generate a HMAC for a message $M$ and then the message and the HMAC value, along with $Apath(K_i)$ and the key are distributed in this format $HMAC(K_i : M) \mid\mid M \mid\mid Apath(K_i) \mid\mid K_i$. The message is transmitted from left to right. The key used for the HMAC is sent in the same message.

TIK eliminates the need for a comparatively long delay between when a message is received and when it can be verified that is the case in the delayed-authentication schemes. However, TIK requires tight synchronization between the sender and receivers and is therefore used by a node to talk to its one-hop neighbors. The synchronization issues are explored in (Hu et al., 2003).

## 4. ANALYSIS

Our focus is to better understand the behavior of authentication mechanisms (in tactical networks) in scenarios relevant to routing protocol security as well as various tactical network relevant applications. We studied three different scenarios: 1) sending messages over multiple hops with end-to-end authentication; 2) sending messages over multiple hops with verification at each hop; and 3) sending an authenticated message to only a node's one-hop neighbors. In each scenario our focus was performance differences between the various schemes during the operational phase (including reinitialization when that is part of normal operations). We also studied the fast startup cost of the various schemes, i.e., combined cost of initializing and sending the first operational message, in these scenarios.

### 4.1 End-to-end authentication operational scenario

We compared the cost of COSP and IOSP schemes (using SHA-1 and MD5 as the hash functions) with the costs of the digital signature schemes from Section 3.1 and Section 3.2. To do so, we used a base message format protected by the different techniques and modeled the incremental cost of each technique. We show the results for 10 hops below. In the first set of graphs we look at relatively high data rate channels, those between 100 Kbps and 10,000 Kbps, see Figure 1, and the second set of graphs show results for low speed channels, 1 Kbps to 100 Kbps, see Figure 2.

The OTDS schemes are more efficient than traditional public-key signature for end-to-end authenti-

[5] Timed Efficient Stream Loss-tolerant Authentication.

[6] The authors discuss other approaches in (Hu et al., 2003).

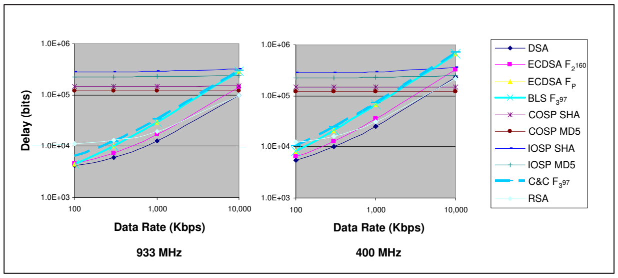

Figure 1: End-to-end authentication - high data rate

cation only in systems with very fast channels, i.e., with data rates near to the 802.11b maximum or the high data rate target for FCS radios (approx. 10,000 Kbps). The LPD mode of FCS radios has a rate of 200 Kbps (Sass and Freebersyser, 2002), and the low data rate mode of the Near Term Digital Radio (NTDR) is 375 Kbps (North et al., 1999). At these rates the OTDS schemes are outperformed by the traditional digital signature schemes for both modes of the Pentium III-M. In order for the OTDS schemes to out perform the traditional signature schemes at these rates; we must reduce the performance of the processor by approximately two orders of magnitude from its maximum.

If special purpose hardware is available for the one-time signature techniques (which would flatten the COSP and IOSP curves at the higher data rates) the impact would be minimal. These curves do not include delay which may be introduced in a system by COSP timing constraints or the cost in ISOP to re-initialize receivers that miss public key updates.

At lower data rates[7] the overhead of the one-time signature schemes is so high that the curves are off the scales used in Figure 2.[8] Here our focus is on the performance of the traditional signature schemes compared to identity-based signatures and short signature schemes based on pairings.

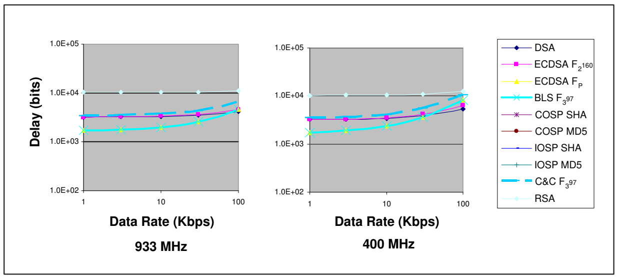

Figure 2: End-to-end authentication - low data rate

The short signature scheme and ECDSA $F_p$ are the preferred choices at these lower rates when the verifier has ready access to the processor in its fast mode. The graphs show that at the lower CPU speed, DSA and ECDSA $F_{2^{160}}$ become completive above 30 Kbps. If we reduce the performance of the processor by two orders of magnitude from its maximum then the DSA is preferred.

### 4.2 Hop-by-hop authentication operational scenario

In this scenario we again compared the cost of COSP and IOSP schemes (using SHA-1 and MD5 as the hash function) with the costs of the digital signature schemes. In this scenario the computational cost of the verification step is magnified and as one might expect, RSA signatures can play a significant role at higher data rates, as shown in Figure 3.

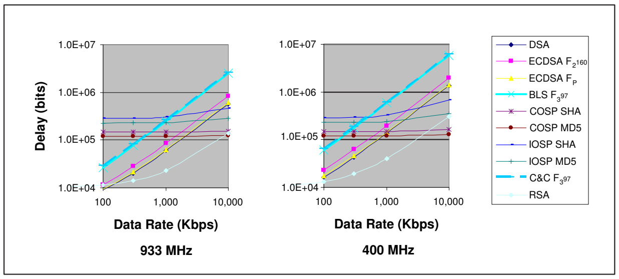

Figure 3: Hop-by-hop authentication - high data rate

In this scenario at the higher data rates, RSA and the OTDS schemes are the main competitors. At the processors' maximum speed RSA is preferred nearly through out the range. With a 400 MHz processor the OTDS schemes begin to out perform RSA at about 3,000 Kbps. If we reduce the performance of the processor by about two orders of magnitude from its maximum then the OTDS schemes outperform the other techniques throughout the higher data rate range.

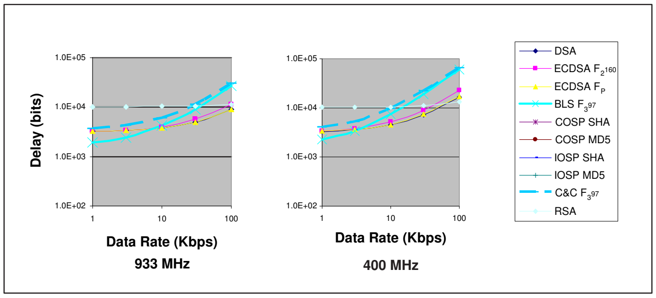

Figure 4: Hop-by-hop authentication - low data rate

At the lower data rates the situation is complex. For the higher speed processor the range is divided, see Figure 4 between the short signature scheme and traditional signature schemes (excluding RSA). As we can see by comparing the two graphs in the figure, this relationship is quite sensitive to small changes in processor

[7] For comparison a number of systems that are part of the Joint Tactical Radio System (JTRS) (JTRS, 2003) have modes with data rates in this range, as well as other systems under development (Nova).

[8] Even without considering packet fragmentation effects which would be unavoidable at such low rates. For example, at 20 kpbs the overhead of COSP would increase the time needed to send a packet over a single hop by about 740 mSec.

performance. If we reduce the performance of the processor significantly then RSA becomes a factor (assuming no fragmentation). If we reduce the performance of the processor by about two orders of magnitude from its maximum then, RSA outperforms the other techniques throughout the lower data rate range.

### 4.3 Single hop authentication operational scenario

We compared the cost of TIK (using SHA-1 and MD5 as the hash function)[9] with the costs of the digital signature schemes from Section 4.1 and Section 4.2. We again used a base message format protected by the different signature techniques and a comparable version of the TIK message, and examined the incremental costs of those techniques.

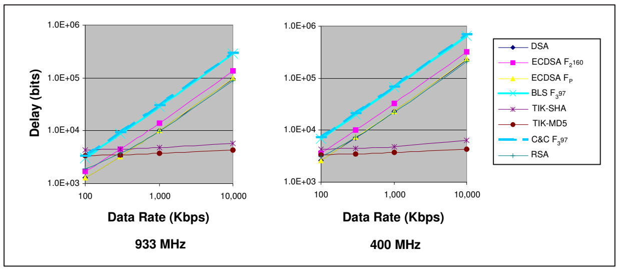

Figure 5: Single hop authentication - high data rate

As before in we look at relatively high speed channels, those between 100 Kbps and 10,000 Kbps in Figure 5, and low speed channels, in Figure 6. In each figure the height of the TIK tree is scaled according to the data rate, using the expression $\lceil 30 - \log_2(data\_rate/10,000Kbps) \rceil$, so that bandwidth is conserved for lower data rate channels. This scaling has the impact of increasing the time interval used by TIK at lower rates. We assume in this analysis that the increase in the interval does not impact delay, e.g., the timing of message generation is matched to the intervals.

The TIK scheme is superior at higher data rates; it is clearly superior at 300 Kbps and higher data rates. If the processor performance is reduced by a third from its maximum then TIK is superior across the entire range. At the lower data rates using the Pentium III-M the short scheme is best at lower rates, up to about 10 Kbps and 5 Kbps for the processor at 933 MHz and 400 MHz respectively. If we reduce the performance of the processor by about one order of magnitude from its maximum, then DSA and ECDSA $F_p$ outperform the other techniques below 30 Kbps. Above 30 Kbps TIK is again superior. If we reduce the performance of the processor by about two orders of magnitude from its maximum then the superiority of TIK extends down to about 3 Kbps.

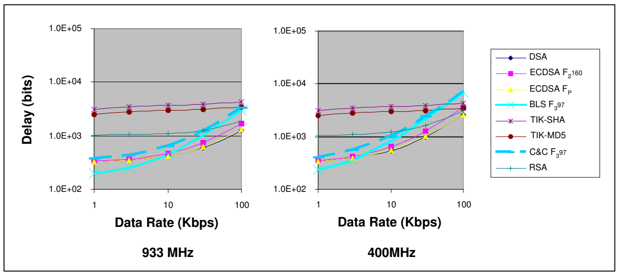

Figure 6: Single hop authentication - low data rate

### 4.4 Fast-startup scenarios

In some situations a critical performance characteristic is how quickly the first authenticated message of an operation phase can be distributed and verified without a prior initialization of the sender to the receivers. In effect, part of the initialization phase is combined with a message from the operational phase.

For each of the traditional signature schemes we increase the size of an operational message by the size of a public key plus a certificate authority's signature in the same traditional signature scheme, and a small amount of additional data. The verifiers' computational cost is increased by an additional signature verification computation. For the identity-based schemes there is no change in message size or computational costs.

For the OTDS schemes the public key of the sender is signed (using ECDSA $F_p$ or Cha and Cheon signatures) by the sender, and the public key signature (and the certificate if necessary), are distributed along with the operational message. The increased computational cost for the signer is from the public key generation process (maybe), and the signing of the public key. For the verifier the increased computational cost is from the signature verification set using ECDSA $F_{2^{160}}$ or Cha and Cheon signatures.

In some scenarios the generation of the public key in an OTDS scheme or an authentication tree will been done in advance. To maximize performance the signer will have the choice of 1) signing its OTDS public key or root node value in advance (using an algorithm from Sections 3.1 or 3.2) and using its OTDS or TIK scheme to authenticate the first message (the *big_comms* mode), or 2) signing the message and the OTDS public key or root node value using an algorithm from Sections 3.1 or 3.2 on the fly (the *small_comms* mode). The second approach has the advantages of lower communication costs and slightly lower verification cost at the price of higher signature generation costs since the computationally expensive signature is not generated in advance. In the

[9] If a an 80 hash were used (as mentioned in (Hu et al., 2003)) the curve would appear in the graphs slightly below the MD5 TIK curve.

remainder of this section we describe the result of our analysis of these forms of fast-startup in the scenarios of Sections 4.1 and 4.2.

### 4.4.1 End-to-end authentication

In all performance ranges we examined the ISOP technique in the *small_comms* mode significantly outperforms ISOP in the it big_comms mode. ISOP in the it small_comms mode also outperforms COSP when used in either mode, by an order of magnitude at about 100 Kbps, and by a factor of 3 at 10,000 Kbps.

We found relatively little difference between using ECDSA $F_{2^{160}}$ or Cha and Cheon signatures to bootstrap the OTDS schemes. We compared differences between the OTDS variants and the various flavors of digital signature schemes in the end-to-end scenario. Figure 7 shows the performance of the various techniques (using Cha and Cheon signatures with the OTDS schemes) using the PIII-M processor.

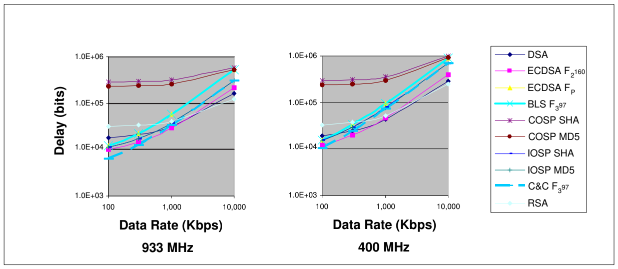

Figure 7: End-to-end authentication - high data rate

The choice of scheme is very sensitive for both processor performance and data rate. At the higher clock speed (933 MHz) RSA is preferred at higher data rates with ECDSA $F_p$ in the middle of the range, and near 100 Kbps Cha and Cheon signatures become competitive. At the slower clock speed (400 MHz) four different schemes have the best performance. From 100 Kbps to 10,000 Kbps we have Cha and Cheon signatures followed by ECDSA $F_{2^{160}}$, then DSA, and finally RSA. IOSP does fairly well across the range, however, the signatures schemes outperform it. Figure 8 shows the performance of the various techniques (using Cha and Cheon signatures with the OTDS schemes) using the PIII-M processor at lower data rates.[10] In this range the Cha and Cheon signatures dominate, especially on the faster processor clock speed.

If we reduce the performance of the 400 MHz processor by a factor of 10 then ECDSA $F_{2^{160}}$ out performs all other schemes at or above approximately 10 Kbps, Cha and Cheon signatures continue to perform best below 10 Kbps.

### 4.4.2 Hop-by-hop authentication

In this scenario the ISOP technique in the *small_comms* mode once again significantly outperforms ISOP in the *big_comms* mode. ISOP in the *small_comms* mode continues to outperform COSP when used in either mode, by an order of magnitude at about 100 Kbps, and by a factor of 3 at 10,000 Kbps. We again found relatively little difference between using ECDSA $F_{2^{160}}$ or Cha and Cheon signatures to bootstrap the OTDS schemes compared with the differences the schemes. Figure 9 shows the performance of the various techniques (using Cha and Cheon signatures with the OTDS schemes) using the PIII-M processor in the higher data rate range.

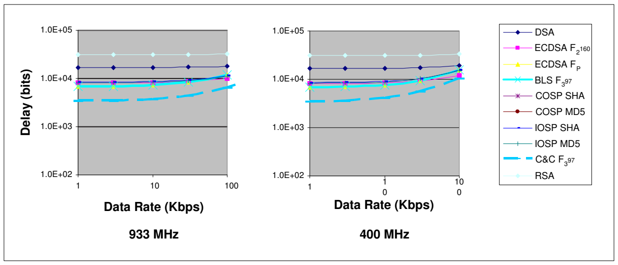

Figure 8: End-to-end authentication - low data rate

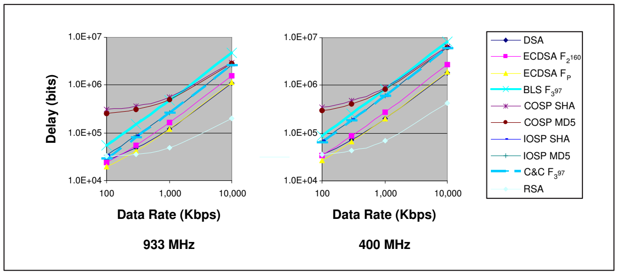

Figure 9: Hop-by-hop authentication - high data rate

On the Pentium III-M at both clock speeds RSA is preferred throughout the entire range. If we reduce the performance by a factor of 10, RSA continues to out perform the other schemes. With such a processor using RSA to initialize an OTDS scheme may be attractive; however, we have not studied this combination. At both clock speeds ECDSA $F_p$ out performs the other techniques on the lower part of the low data rate range. Figure 10 shows the performance of the various techniques (using Cha and Cheon signatures with the OTDS schemes) using the PIII-M processor in the lower data rate range.

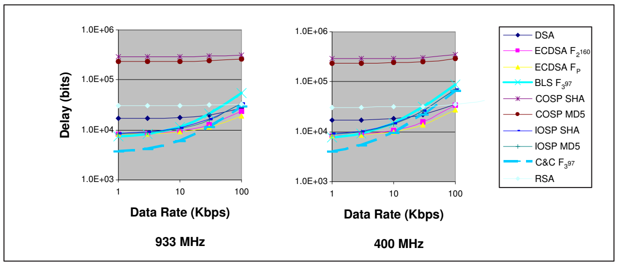

Figure 10: Hop-by-hop authentication - low data rate

At the higher clock speed ECDSA $F_p$ dominates

[10] The COSP scheme is off the scale.

above 30Kbps while Cha and Cheon signatures do the best below 30 Kbps. At the lower clock speed (400 MHz) ECDSA $F_{2^{160}}$ is preferred above approximately 15 Kbps and Cha and Cheon signatures do the best below that range. If we reduce the performance of the 400 MHz CPU by an order of magnitude (not shown), then RSA is preferred above approximately 20 Kbps and ECDSA $F_{2^{160}}$ is preferred below.

## 5. CONCLUSIONS AND FUTURE WORK

In order to understand the relative cost of various authentication techniques in tactical networks we used a simple model to compare traditional and new digital signature techniques against recently developed, novel, authentication techniques using one-time signature and authentication trees. Our results show that over the wide range of performance exhibited by tactical systems and interesting scenarios, no specific technique provides the best performance. The TIK technique, when applicable, performs very well across a wide range of data rates and processor capabilities. The high verification cost of identity-based signatures and pairing-based short signatures limits their use to lower bandwidth channels in the non fast startup scenarios. When fast startup is needed or the application needs self-contained messages, i.e., all the information (other than system parameters) needed to authenticate messages is distributed with the messages. Identity-based signatures become competitive for end-to-end authentication, and to a limited extent for hop-by-hop authentication.

Further esearch is needed understand the impact of fragmentation and other characteristics of CSMA channels, as well as other channel access techniques, will have on the performance of these authentication techniques. Research is also needed in other performance parameters, e.g., energy consumption, as well as other interesting authentication schemes, including other identity-based signature schemes and other broadcast authentication protocols (Perrig et al., 2001; Reyzin and Reyzin, 2002).

## REFERENCES

Barreto P., Kim H. Y., Lynn B., and Scott M., Effcient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – Crypto'2002, volume 2442 of Lecture Notes in Computer Science*, 2002.

Barreto P., *Criptografia Robusta e Marcas d'gua Frgeis: Construo e Anlise de Algoritmos para Localizar Alteraes em Imagens Digitais*. PhD thesis, Universidade de So Paulo, Escola Politcnica, 2003.

Boneh D., Lynn B., and Shacham H., Short signatures from the Weil pairing. In *Proceedings of Asiacrypt 2001, volume 2248 of LNCS*, 2001.

Cha J. and Cheon J., An identity-based signature from gap diffie-hellman groups. In *PKC'2003, Lecture Notes on Computer Science 2567*, 2003.

FIPS 186-2, U.S. Department of Commerce N.I.S.T., *FIPS 186-2, Digital signature standard, Federal Information Processing Standards Publication 186-2, Change Notice 1*, 2001.

Hauser R., Przygenda A., and Tsudik G., Reducing the cost of security in link state routing protocols. In *ISOC Symposium on Network and Distributed Systems Security*, 1997.

Hu Y., Perrig A., and Johnson D., Packet leashes: A defense against wormhole attacks in wireless networks. In *IEEE Infocom 2003*, April 2003.

JTRS, *Joint Tactical Radio System (JTRS) Operational Requirements Document (ORD)*, (Extract of JROC approved final with waveform Table 4-2 and ANNEX E), Version 3.2 JROC Approved, JROCM 087-039, April 2003.

Lamport L., Constructing digital signatures from a one way function. Technical report, Technical Report CSL-98, SRI International, 1979.

Merkle R., Protocols for public key cryptosystems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1980.

North R., Bryan C., and Baker D., Wireless networked radios: Comparison of military, commercial and R&D proposals. In *Proceedings of the 2nd Annual UCSD Conference on Wireless Communications*, 1999.

Nova Engineering website, http://www.nova-eng.com.

Perrig A., The BIBA one-time signature and broadcast authentication protocol. In *ACM Conference on Computer and Communications Security*, 2001.

Perrig A., Canetti R., Tygar D., and Song D., The TESLA broadcast authentication protocol, Cryptobytes, Volume 5, No. 2 (RSA Laboratories, Summer/Fall), 2002.

Reyzin L. and Reyzin N., Better than biba: Short onetime signatures with fast signing and verifying. In *Seventh Australasian Conference on Information Security and Privacy (ACISP)*, 2002.

Rivest R., Shamir A., and Adleman L., A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 1978.

Sass P. and Freebersyser J., FCS communications technology for the objective force. Technical report, MITRE Technical Report, 2002.

Shamir A., Identity-based cryptosystems and signature schemes. In *Proc of Crypto'84*, pages 47–53, 1985.

Zhang K., Efficient protocols for signing routing messages. In *Symposium on Network and Distributed Systems Security (NDSS'98)*, January 1998.